Supplemental Document for Front2Back: Single View 3D Shape Reconstruction via Front to Back Prediction

Yuan Yao[1] Nico Schertler[1] Enrique Rosales[1, 2] Helge Rhodin[1] Leonid Sigal[1, 3] Alla Sheffer[1]
[1]University of British Columbia [2]Universidad Panamericana
[3]Canada CIFAR AI Chair, Vector Institute
{rozentil, nschertl, albertr, rhodin, lsigal, sheffa}@cs.ubc.ca

In this document, we supply additional evaluation, training, and implementation details, and provide a more detailed ablation study.

1. Additional qualitative results

We included only few qualitative experiments in the main paper due to space limitation. In Figure 1 we provide additional experiments that illustrate performance on a broader set of classes and compare our results against state-of-the-art alternatives. In all cases we are able to recover more faithful topology and geometry of objects, including local high-fidelity detail. Our results consistently more accurately reflect the ground-truth geometry. Among all methods compared against, we come closest in approximating the geometry of the lamps (rows 1, 6), and capturing the details of the boats (rows 4,5) and the airplane (row 7). Contrary to Pixel2Mesh [9] we correctly recover the topology of the input models, accurately capturing handles on the display (row 2) and bed (row 3) and correctly recovering the chair back (rows 8, 9) and table (last row) leg connectivity. Contrary to methods such as OccNet [8] and IM-NET [2] we do not hallucinate non-existent details (see car, row 10, gun row 11, or table, last row).

2. Additional quantitative evaluation

Due to lack of space, in the main paper we only report Chamfer L_1 distance (CD) measured using the implementation in [8] and surface-to-surface distance (MD), as computed by Metro [4], as our error metrics. Here we include additional evaluation, with respect to state-of-the-art, based on the Normal Consistency metric introduced by [8]. Results are reported in Table 1. Since we measure consistency, higher numbers indicate better performance, with 1 being the optimum. Our proposed method works better on 8 out of 13 categories and overall. This provides additional evidence that we improve on the previous state-of-the-art performance. Since we measure accuracy directly against unprocessed ShapeNet models, most of which are not watertight, we do not provide IoU measurements, since those are only valid for comparing watertight meshes.

Additional information on experiments. When comparing our results to other methods we encountered two challenges. First, a number of methods, *e.g*. OccNet [8] and IM-NET [2] that use watertight proxies of ShapeNet models for training, measure result accuracy against these proxies, instead of the originating ShapeNet models. We seek to measure accuracy against the original ShapeNet models. Second, while we followed the original ShapeNet train/test/val split to evaluate our models, other methods *e.g*. Pixel2Mesh [9], AtlasNet [5] use the split provided by [3]. To resolve both issues we use pre-trained models provided by the authors of the relevant papers and test them on the intersection of the two test sets. This intersection test set contains 1685 models: 197 airplanes, 83 benches, 62 cabinets, 149 cars, 324 chairs, 48 displays, 101 lamps, 63 speakers, 75 rifles, 132 sofas, 339 tables, 46 cellphones, and 66 boats. We then measure all reported metrics on these results, comparing them against original ground truth ShapeNet [1] models.

Each method we compare to uses a different scaling convention. We apply the transformations used by those methods to transform their outputs accordingly to match ground truth models. For AtlasNet [5], OccNet [8] and Pixel2Mesh [9], we do the exact transformations they described in their papers and supplementary materials. For IM-NET [2] we follow the transformations used in the voxelization process they utilized [6].

3. Scalability to High Resolution

Our Front2Back map-based formulation can easily scale to operate at the higher resolutions. To do so, we simply need to train (i) image to front map and (ii) front2back modules to operate at the higher resolution (*i.e*., take higher resolution image as input and produce, equivalent, higher resolution depth and normal maps as outputs). The rest of our framework can remain exactly as is. To illustrate this beneficial capability, we report additional experiments we didn't

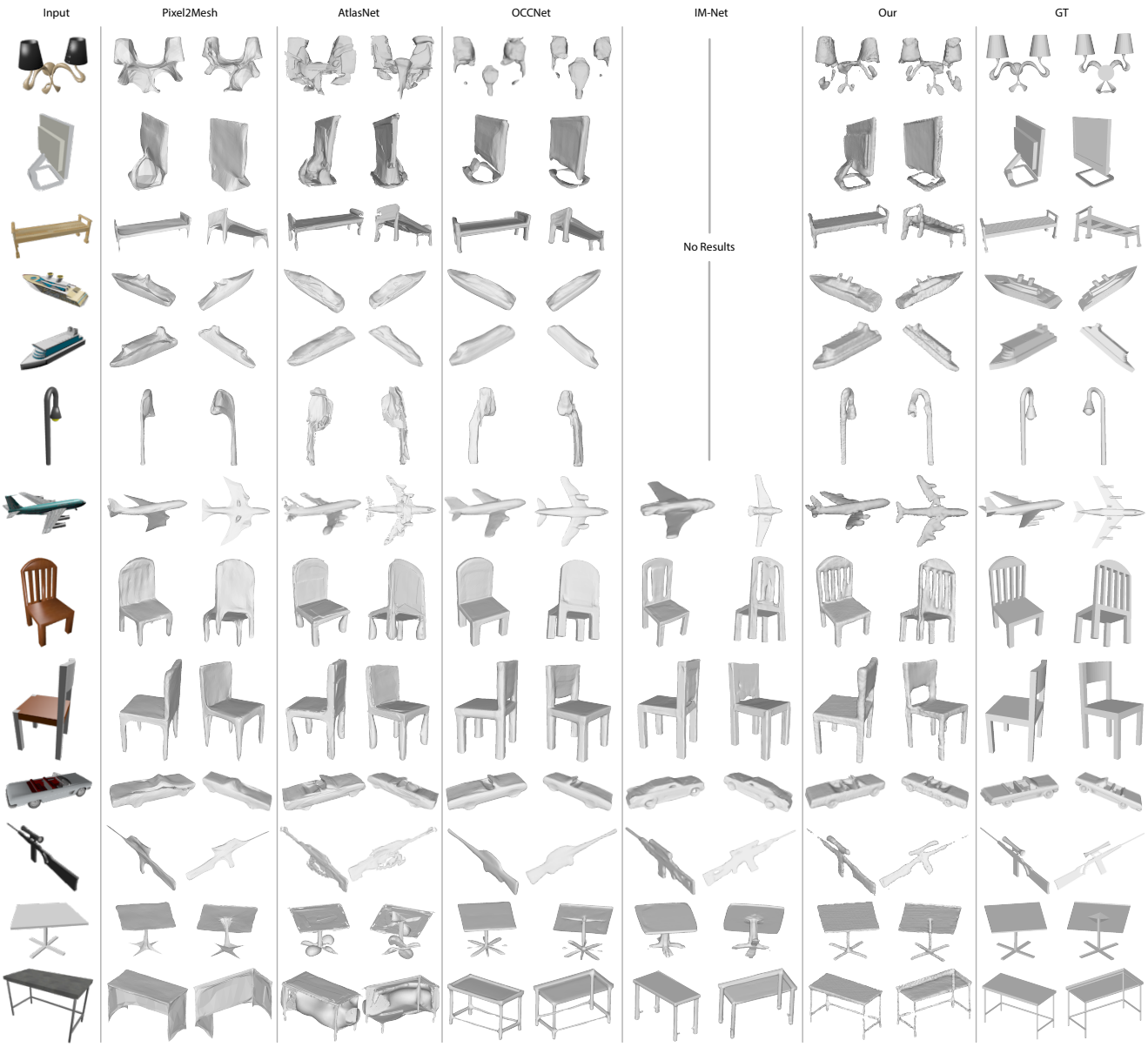

Figure 1: **Qualitative comparison to state-of-the-art.** Visual comparison of our results and those produced by Pixel2Mesh [9], AtlasNet [5], OccNet [8], and IM-NET [2]. Our results can recover the shape of the input well and capture much more details than others, without halucinating non-existent details. Note that as IM-NET [2] only provides pre-trained models on five classes, we cannot compare with them on other classes such as lamps, displays, etc.

have space to include in the main paper. We report performance at 256×256 resolution on the `chair` and `plane` classes. Results are qualitatively illustrated in Figure 2 and quantitatively analyzed in Table 2. Higher resolution maps allow recovery of finer geometric detail observed in the outputs. For example, in our higher-resolution results, the legs on the chairs appear much cleaner and geometrically more accurate (row 2) and planes have much more decernable engines and other fine detail (row 3 and 4). This is quantified in Table 2, which shows significantly improved reconstruction scores (approximately 15% improvement on average) as compared to our main 137×137 resolution model result.

4. Implementation Details

In this section, we add more details for our implementation which are not listed in the main paper.

Symmetry plane detection. We represent the symmetry planes as 3-dimensional vectors (ϕ, θ, d), where ϕ, θ are the polar angles of the plane's orientation, and d is the distance

Methods	Category													Mean
	chair	plane	car	bench	cabinet	display	lamp	speaker	rifle	sofa	table	phone	vessel	
Ours	**0.771**	**0.759**	0.734	0.674	0.763	0.821	**0.734**	**0.780**	**0.672**	**0.781**	**0.789**	0.856	**0.733**	**0.759**
ONet (Mescheder *et al.* [8])	0.741	0.719	**0.759**	0.668	0.759	0.771	0.707	0.729	0.590	0.757	0.725	0.824	0.667	0.724
AtlasNet (Groueix *et al.* [5])	0.701	0.682	0.725	0.627	0.725	0.772	0.641	0.738	0.549	0.734	0.671	0.839	0.614	0.694
Pixel2Mesh (Wang *et al.* [9])	0.741	0.747	0.698	**0.685**	**0.782**	**0.837**	0.705	0.767	0.636	0.780	0.762	**0.881**	0.681	0.746
IM-NET (Chen *et al.* [2])	0.686	0.687	0.680	/	/	/	/	/	0.579	/	0.657	/	/	0.658

Table 1: **Normal Consistency [8] comparisons against state-of-the-art.** We compare our results against Pixel2Mesh [9], AtlasNet [5], OccNet [8], and IM-NET [2] measuring Normal Consistency (larger value better). Our method provides the best results overall and outperforms the closest competitors on 8 out of 13 classes.

	Methods	chair	plane
MD	Ours - 137	0.013	0.013
	Ours - 256	**0.011**	**0.011**
CD	Ours - 137	0.021	0.017
	Ours - 256	**0.018**	**0.015**

Table 2: Quantitative results on high resolution (256×256) inputs versus low resolution (137×137) inputs. The performance is increased by 10%-20%.

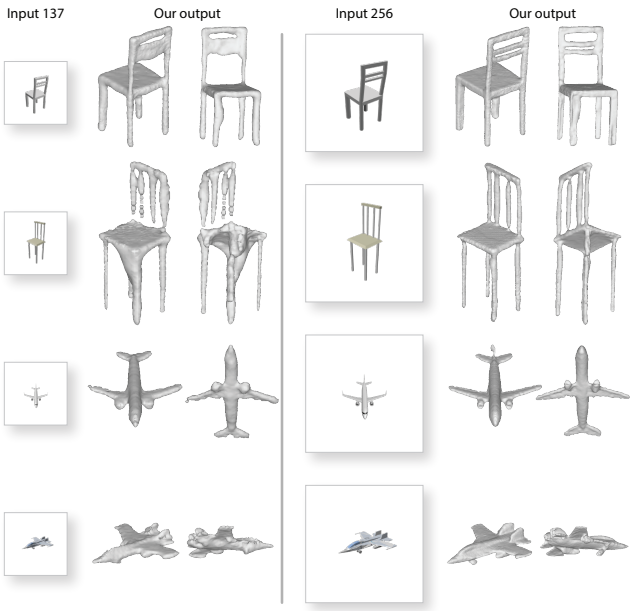

Figure 2: 137×137 **vs** 256×256 **resolution results.** (left) low resolution images and results. (right) similar view and render style 256×256 images and corresponding results. The quality of our reconstruction predictably improves with increase in image resolution.

of the plane to the origin. We use mean-shift clustering with the Epanechnikov kernel to generate an initial symmetry plane guess.

Neural network architecture. We train pix2pix [7] with the resnet-9 block architecture to predict the back view oriented depth maps. Our input images are 256×256 pixels and have eight channels (three channels for the normals, one channel for the depth; both for front view and reflected view). In case of 137×137 model we pad to 256; for the high-resolution model we directly use 256×256 inputs. For the front normal and depth map prediction we train separate networks for 137×137 and 256×256 resolutions.

Training scheme. We train separate models for the image to 2.5D map representation and the back2front reconstruction. Both networks are trained independently on ground truth data. The learning rate is set to $2e-4$ and decays by 0.5 after 20 epochs; we train for a total of 40 epochs. Our training data set contains independent classes such as car, airplane, *etc.*; we train separate models for each class. This is consistent with other papers in the area.

5. Limitations

Our method performs well on typical views of everyday objects, where the front and back views (combined with the reflection of the front) describe a big portion of the shape. Predictably, its performance declines on accidental views where the union of these information sources is insufficient to describe the surface (see Figure 3a). A possible solution for such scenarios is to perform surface completion, *e.g.* using the method of [8]; or to leverage our existing algorithm rendering the incomplete output from a different direction (ideally one maximally opposite to the incomplete data); repeating the back prediction step, and performing reconstruction using a union of previous and new maps. Our framework does not hallucinate geometry not evident from the images, thus given fuzzy, ambiguous images it is not able to produce meaningful reconstructions (Figure 3b). Symmetry computation on raw images or predicted front maps is a challenging problem; incorrect symmetry plane estimation can potentially lead to reconstruction failures (see Figure 3c). We minimize the likelihood of such failures by using very conservative symmetry detection criteria; more robust detection of symmetry planes on image inputs is an interesting future research topic.

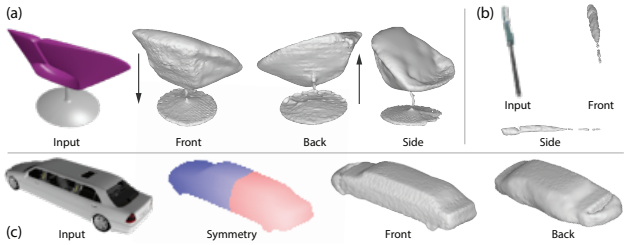

Figure 3: **Limitations.** (a) Our method's performance declines on accidental view images, such as these, where important details are not discernible from front or predicted back views. (b) Our method cannot recover fuzzy details from ambiguous low-resolution inputs. (c) A wrongly detected reflective symmetry (here left/right instead of front/back) can lead to reconstruction artifacts.

References

[1] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], 2015. 1

[2] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 2, 3

[3] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *The European Conference on Computer Vision (ECCV)*, 2016. 1

[4] Paolo Cignoni, Claudio Rocchini, and Roberto Scopigno. Metro: measuring error on simplified surfaces. In *Computer Graphics Forum (CGF)*, volume 17, 1998. 1

[5] Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan C. Russell, and Mathieu Aubry. A papier-mâché approach to learning 3d surface generation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 2, 3

[6] Christian Häne, Shubham Tulsiani, and Jitendra Malik. Hierarchical surface prediction for 3d object reconstruction. In *2017 International Conference on 3D Vision (3DV)*, pages 412–420. IEEE, 2017. 1

[7] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 3

[8] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 2, 3

[9] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. In *The European Conference on Computer Vision (ECCV)*, 2018. 1, 2, 3